

4-28-2014

Solar Tracking System

Mohamad El Atwani

Indiana University - Purdue University Fort Wayne

Patrick Koepper

Indiana University - Purdue University Fort Wayne

Follow this and additional works at: http://opus.ipfw.edu/etcs_seniorproj_engineering



Part of the [Mechanical Engineering Commons](#)

Opus Citation

Mohamad El Atwani and Patrick Koepper (2014). Solar Tracking System.
http://opus.ipfw.edu/etcs_seniorproj_engineering/58

This Senior Design Project is brought to you for free and open access by the School of Engineering, Technology and Computer Science Design Projects at Opus: Research & Creativity at IPFW. It has been accepted for inclusion in Engineering Senior Design Projects by an authorized administrator of Opus: Research & Creativity at IPFW. For more information, please contact admin@lib.ipfw.edu.

Indiana University-Purdue University Fort Wayne
Department of Engineering

ECE 405 - ECE 406
Capstone Senior Design Project
Report # 2

Project Title: Solar Tracking System

Group Members: Mohamad El Atwani
Patrick Koepper

Faculty Advisor: Dr. Hossein Olbomi

Date: April 28th, 2014

Table of Contents

Index of Figures and Tables	3
Acknowledgements	4
Abstract	5
Section I – Detailed Design Description	6
1.1 Initial Conceptual Design.....	7
1.2 Requirements and Specifications	8
1.3 Given parameters	8
1.4 Design variables	8
1.5 Limitations and constraints	8
1.6 Safety, environmental, economic, and other considerations	9
1.7 Revisions on Initial Design	9
1.8 Final Prototype Design	9
Section II - Prototype Construction	11
2.1 Prototype Construction	12
2.1.1 Building the Circuits	12
2.1.2 Frame Assembly	13
2.1.3 Solar Panel Modification	15
2.1.4 Motor and Pulley Installation	16
2.1.5 Microcontroller Software Development	18
2.2 Prototype Final Cost	20
2.2.1 Framing Material	20
2.2.2 Solar Panel and Modifications	20
2.2.3 Electrical Components	20
2.2.4 Overall Cost	20
2.3 Completed Prototype	21
Section III – Testing	22
3.1 Testing Requirements and Methods	23
3.1.1 Angular Speed of Solar Platform.....	23
3.1.2 Power Consumption	23
3.1.3 System Accuracy	24
3.1.4 Prototype Weight	25
3.1.5 Power Output Improvement	25
Section IV – Conclusions and Recommendations	28
4.1 Conclusions.....	29
4.2 Recommendations for Future Designs	29
Appendix A - Source Code for Arduino	30
Appendix B - Detailed Bill of Materials	35
Appendix C - Raw Solar Panel Output Data	36

Index of Figures and Tables

Figure 1 - Detailed Design of the Platform Frame	7
Figure 2 - Circuit Diagram and Pin connections for Initial Conceptual Design	7
Figure 3 - Completed Prototype Side 1.....	10
Figure 4 - Completed Prototype Side 2.....	10
Figure 5 - Final Prototype Circuit Diagram.....	12
Figure 6 - Control system circuitry.....	13
Figure 7 - 80/20 L-bracket attaching Frame Members	13
Figure 8 - 80/20 Extended Bracket Attaching Frame Members	14
Figure 9 - Pipe Strap Securing Drive Shaft Bearing.....	14
Figure 10 - 10mm Bore Clamping Hub	15
Figure 11 - Top view of solar panel with photoresistors installed.	15
Figure 12 - Detailed View of Installed photoresistor.....	16
Figure 13 - Detailed circuit diagram of a DC motor attached to the platform.....	16
Figure 14 - Motor and Pulley Setup Side View	17
Figure 15 - Motor and Pulley Setup Front View	17
Figure 16 - Microcontroller code flow chart	19
Figure 17- Prototype Completely Constructed	21
Figure 18 - Ammeter Display of Current Draw under Maximum Load	24
Figure 19 - Ammeter Display of Current Draw when System is Idle	24
Figure 20 - Current Sensor in Control Circuitry.....	25
Figure 21 - Solar Panel Output of Tracking System (Light at 16 Degrees).....	26
Figure 22 - Solar Panel Output of Tracking System (Light at 45 Degrees).....	26
Figure 23 - Solar Panel Output of Tracking System (Light at 90 Degrees).....	27
Table 1 - Overall Cost of the Prototype Materials	20
Table 2 - Detailed Bill of Materials	35
Table 3 - Data Comparison between Tracking and Stationary System (Light at 16 Degrees).....	36
Table 4 - Data Comparison between Tracking and Stationary System (Light at 45 Degrees).....	36
Table 5 - Data Comparison between Tracking and Stationary System (Light at 90 Degrees).....	37

Acknowledgements

The team would like to thank the IPFW Department of Engineering for funding this project and also providing us with the knowledge required to puzzle through and complete the difficult sections of the design.

We would also like to thank our Faculty Advisor, Dr. Oloomi, for helping us organize our ideas and providing useful guidance in the design and implementation of the control system for the project. His expertise and advice has led us through the journey of developing the design for this project. Additionally, the team would like to thank Dr. Donald Mueller for providing mechanical insight into the design of the project and supplying the solar panel.

Abstract

Stationary solar energy systems provide less than their optimum power output during many hours of operation. This is due to the fact that solar panels provide their maximum output power when the panel is perpendicular to the sun's light, which is only for a short period of the day when the panel is stationary. Solar tracking systems resolve this problem and have been around for many years. Some systems use light sensor arrays to track the light intensity of the sun, while others use mathematical models that have no external sensors and just set the angle of the panel at certain times during the day.

The engineering design process of this project is divided into two semesters, where the design process performed during the first semester has already been outlined in the first report. The intent of this document is to illustrate the building, testing and analysis of a lower cost solar tracking system primary design chosen to best address the problem statement. This report outlines the completed design from last semester, the building process and design changes done prior to testing, all tests completed, and a full evaluation of the design.

Based on the results and evaluations of the completed tests, the design meets all of the specified requirements that were discussed with the advisor. In addition to that, recommendations for a better design or changes that may be implemented in the future are also detailed in the report.

Section I – Detailed Design Description

1.1 Initial Conceptual Design

During the last semester, a final conceptual design primarily using two photoresistors, accelerometer, microcontroller, DC motor, user interface (for angle input), and platform with solar panel would be sufficient to meet all of the design requirements and specifications. The final conceptual design can be seen in Figures 1 and 2 below.

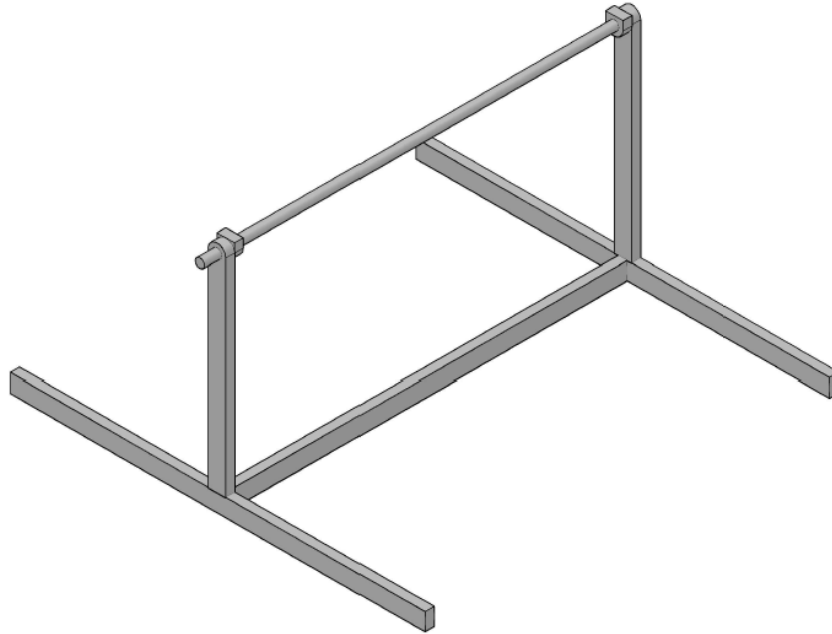


Figure 1 - Detailed Design of the Platform Frame

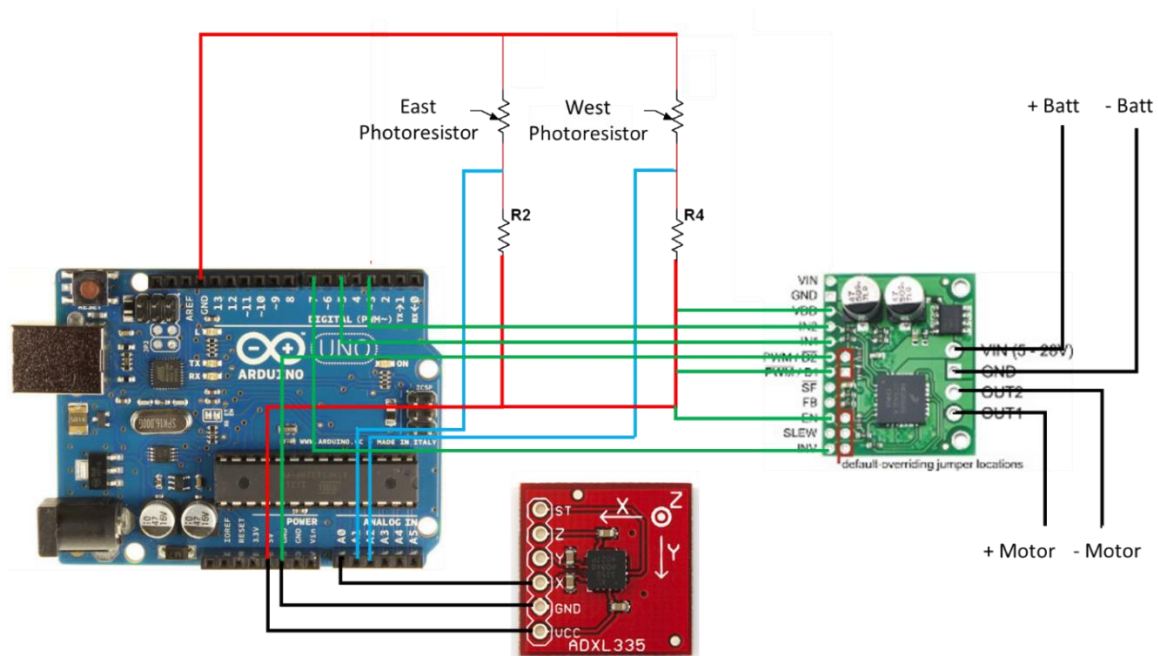


Figure 2 - Circuit Diagram and Pin connections for Initial Conceptual Design

1.2 Requirements and Specifications

- Timing - The platform must be able to pivot to the point with the most solar intensity, from any position, in less than 60 seconds.
- If the solar panel doesn't detect a certain threshold of light intensity (at least enough to produce 25% of the solar panels max output) it must enter power saving mode.
- Power – The control circuitry must be able to consume less than 25% of the onboard battery during time of low light (during power saving mode) during a full day.
- The control system should continuously check the position of the sun so that the panel is always perpendicular to the sun in order to receive the maximum amount of light.

1.3 Given parameters

- The platform frame and circuitry should weigh less than 20lbs, not including the solar panel itself.
- Single Axis Solar Tracker – Must have a rotational axis between 20 degrees and 160 degrees from the panel platform.
- Energy source – all components must be able to run off of a single 12Volt battery.

1.4 Design variables

- Control Unit – The control unit conducts the sun tracking information and coordinates the movement of the positioning system.
- Position Sensor – The position sensor can be changed in order to meet the specific accuracy and power requirement of the system.

1.5 Limitations and constraints

- Cost – The allotted budget of this project from by the IPFW Department of Engineering was initially set at \$200. The budget was raised to \$400 during the second semester of Senior Design.
- Time – The group has until the end of the second semester to finish the conceptual design and implementation.
- The platform must be accurate to within the nearest 3 degrees or 0.052 radians. The requirement was changed from the first semester limitations based on research into solar panels which said that a misalignment of 8° typically results in a 1% power reduction in the panel.

1.6 Safety, environmental, economic, and other considerations

- The circuitry must also be able to with rainy conditions.
- Controlling the panel on a dual axis rotation instead of just one that is mainly considered.
- A conditional goal, if time permits if, of the project is to send the status (solar panel output, current platform position, and controller power consumption) of the system to a remote device.

1.7 Revisions on Initial Design

The final design of the prototype had a few changes when compared to the final design of the first semester's conceptual design. The conceptual design had a mode which allowed the user to input a desired angle and then require the motor to travel to that specific angle. Due to budgetary and time restrictions, this mode could only be implemented partially by changing the desired angle in the Arduino Microcontroller's code (currently set to 90°) and the use of a button that would trigger an external interrupt on the microcontroller in order to change between the accelerometer and light sensor modes. With a little extra time and money, a small LCD and keypad could be implemented to incorporate a full function user input mode as all of the motor control coding has already been implemented to allow it.

The original concept's framing material was initially intended to be fashioned from aluminum bars. These bars were dropped in favor of 80/20 aluminum framing members which are more versatile and easier to assemble/disassemble.

The final prototype also utilized a speed reducing gear, which would allow the DC motor to operate at slower speeds that are more realistic for operations which were tracking the Sun's movements, and LED indicators of which photoresistor was receiving more light.

1.8 Final Prototype Design

The final design varied very little from the initial design concept. The control system included two light sensors (photoresistors), accelerometer, microcontroller, motor driver, 12VDC motor, and 12V motor. These were interfaced with the solar panel and frame structure using a speed reducing gear and pulley. Images of the final design can be seen in Figure 3 and Figure 4 below.



Figure 3 - Completed Prototype Side 1

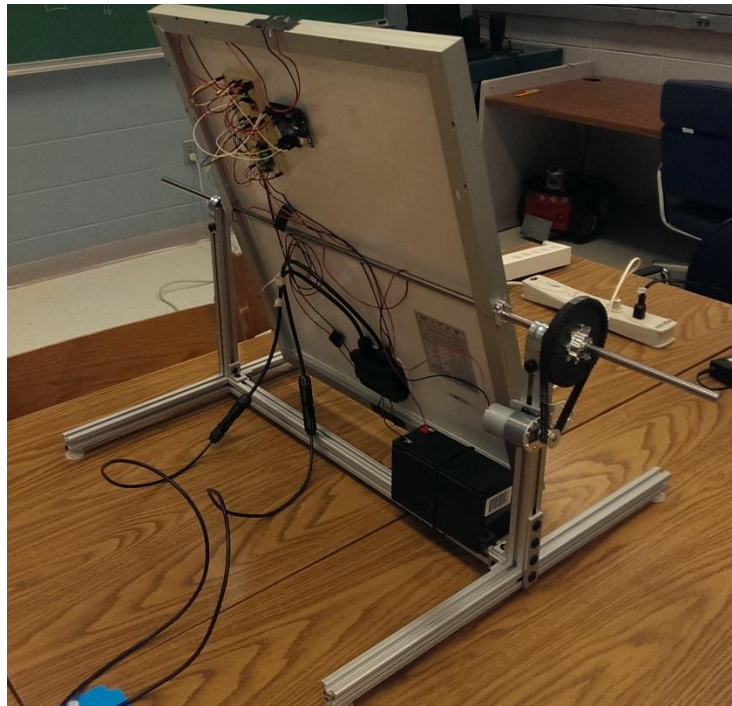


Figure 4 - Completed Prototype Side 2

Section II - Prototype Construction

2.1 Prototype Construction

2.1.1 Building the Circuits

The components of the circuit diagram, outlined in Figure 5 below, are the microcontroller, a motor driver, two light sensors, and an accelerometer. The sensors draw power from the microcontroller's 5V voltage regulated power supply. The Arduino microcontroller then reads the voltage at the voltage divider circuit using the analog inputs. It then sends a scaled PWM (Pulse Width Modulated) signal, using the PWM enabled digital I/O pins, to the motor driver based on the difference in the voltages. The motor driver uses these PWM signals to control the output speed and direction of the DC motor.

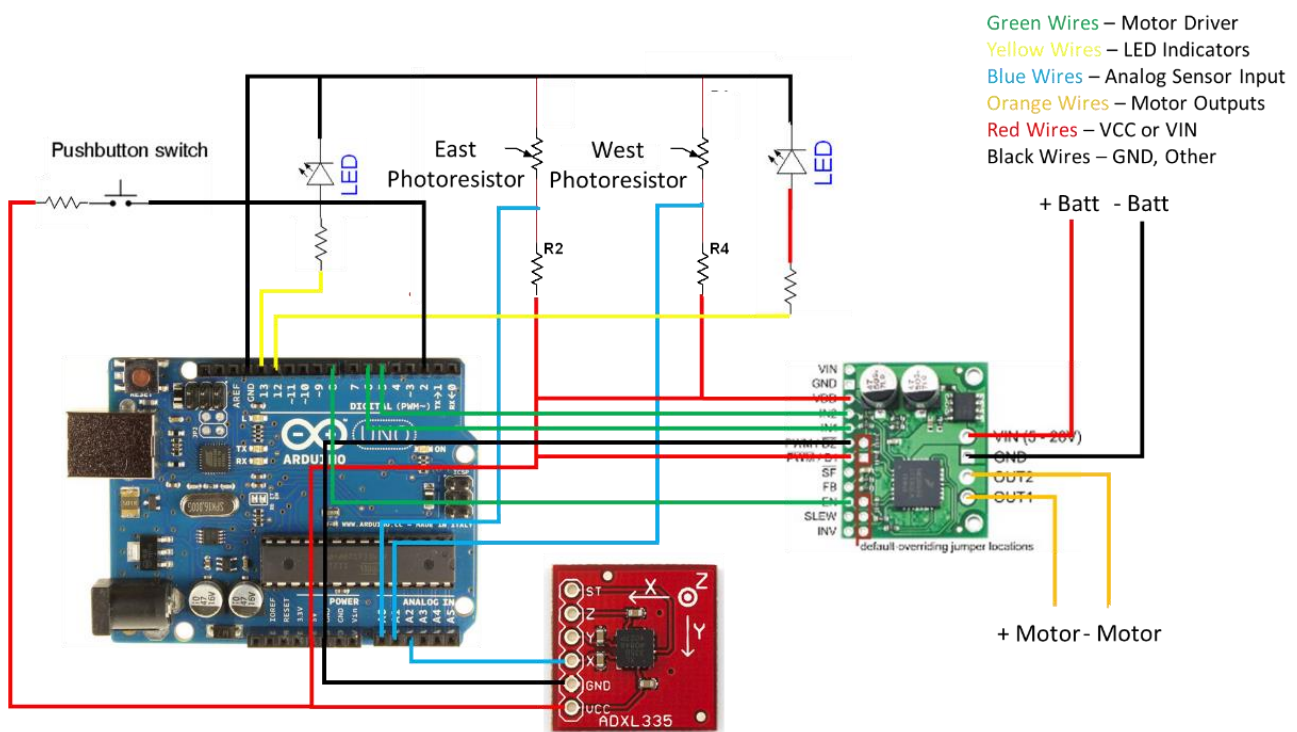


Figure 5 - Final Prototype Circuit Diagram

The circuit diagram was constructed on a breadboard using jumpers to connect the motor driver and accelerometer to the Arduino microcontroller and varying lengths of wires to connect the DC motor and battery supply to the motor driver. Figure 6 shows the completed circuitry and wiring.

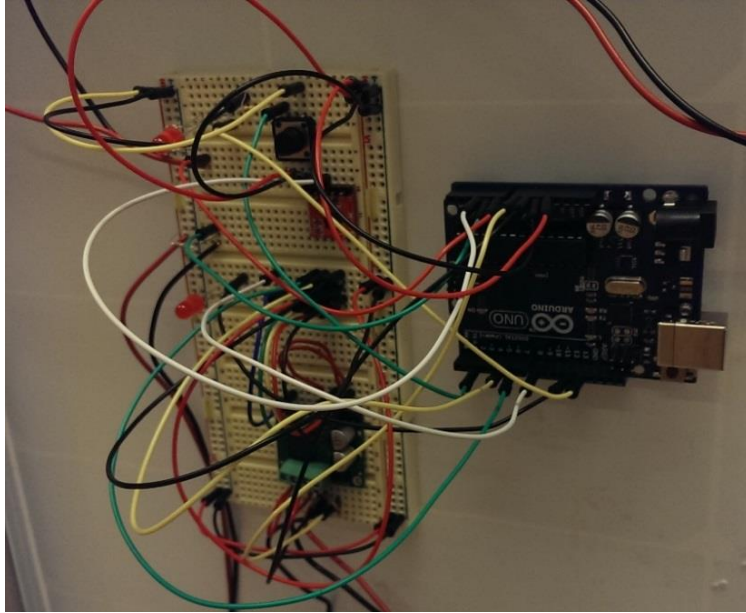


Figure 6 - Control system circuitry

2.1.2 Frame Assembly

The first step of the frame assembly to be completed was of the frame and platform which would house the control circuitry and the solar panel. The framing material is made from 80/20 which is an extruded, slotted aluminum modular framing material. Five total pieces (2 x 1' sections and 1 x 1' sections) of 80/20 were used, as detailed in Figure 1, and connected with 80/20 L-brackets or extended brackets as see in Figure 7 and Figure 8 below. Feet were attached at the four corners of the frame's base.



Figure 7 - 80/20 L-bracket attaching Frame Members

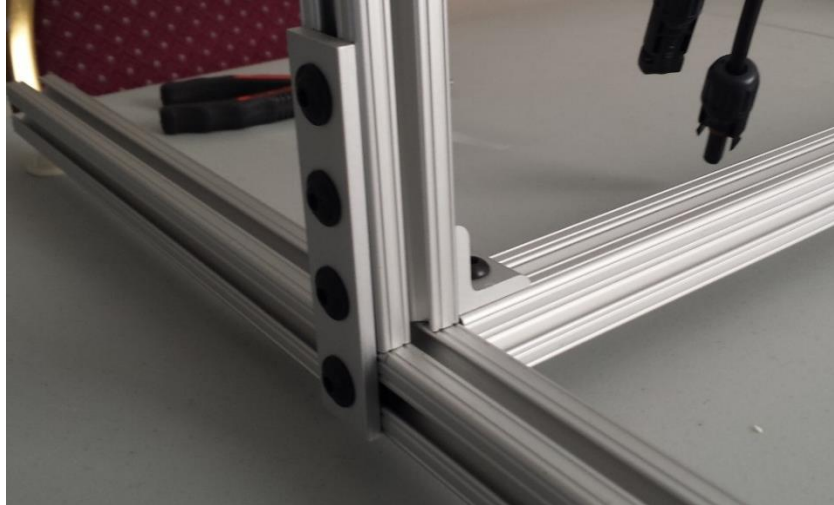


Figure 8 - 80/20 Extended Bracket Attaching Frame Members

The second step was the installation of the bearings and rotary shaft that would support the solar panel. A 10mm aluminum bearing was superglued to the top of each of the vertical 1' shafts of the frame using the 3', 10mm diameter drive shaft as a guide to ensure the correct placement of the bearings. Once the glue had dried completely, a pipe strap was fastened over the bearings and secured to either side of the vertical frame using 80/20 fasteners, as seen in Figure 9 - Pipe Strap Securing Drive Shaft Bearing below.

Finally, a 37mm diameter motor bracket which was secured to the frame with 80/20 fasteners to complete the main frame structure.

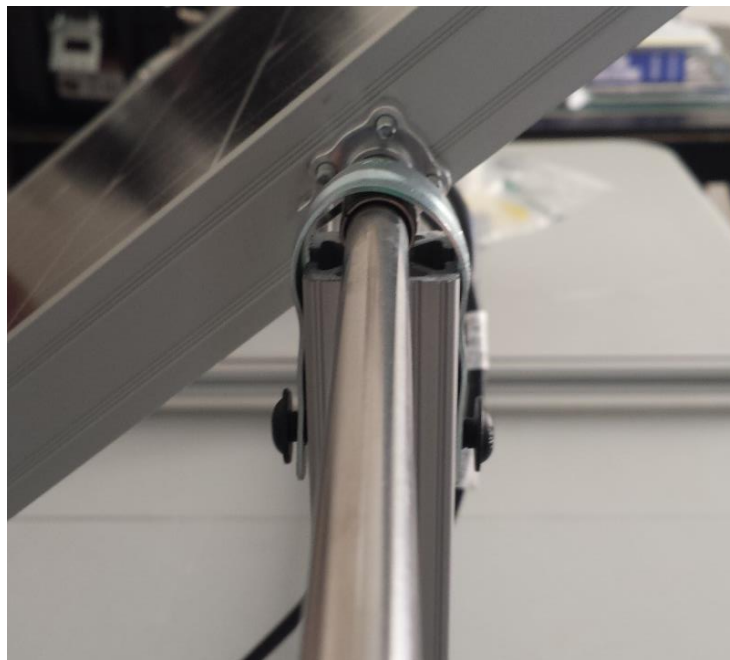


Figure 9 - Pipe Strap Securing Drive Shaft Bearing

2.1.3 Solar Panel Modification

The solar panel frame had to be modified in order to attach its frame to the drive shaft which would hold it onto the frame. An 11mm hole was drilled through the side wall of the solar panel's frame along with four 3mm holes that would allow the installation of a 10mm bore clamping hub (shown in Figure 10) to either side of the solar panel frame. These clamping hubs would allow easy removal of the solar panel from the drive shaft. Figure X shows the completed assembly of the clamping hub to the solar panel frame and drive shaft.



Figure 10 - 10mm Bore Clamping Hub

The two photoresistors were then attached to either end of the solar panel using electrical tape and super glue to permanently secure them to the panel as shown in Figure 11 and Figure 12.

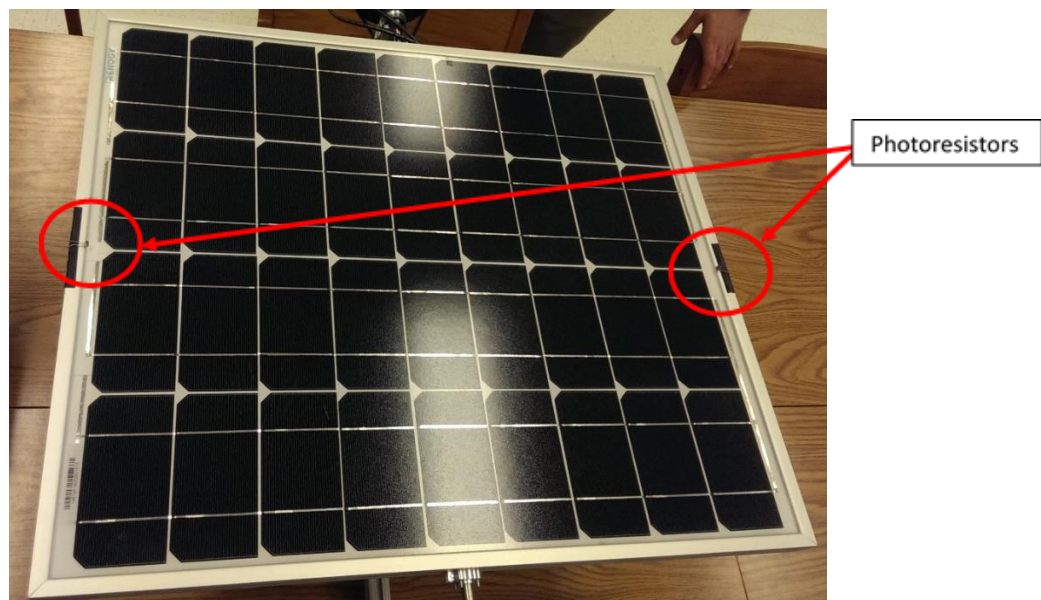


Figure 11 - Top view of solar panel with photoresistors installed.



Figure 12 - Detailed View of Installed Photoresistor

2.1.4 Motor and Pulley Installation

The final portion of the mechanical side of the prototype was the installation of the motor and pulley that is controlled by the microcontroller and control circuitry (discussed later). A 3.8" diameter pulley with 1/5" pitch was attached to the drive shaft using a clamping hub (shown in Figure 10) and adaptors. This pulley was used as a speed reducer from the 12VDC, 80 RPM motor's 0.6" diameter timing pinion.

Figure 13 - Detailed circuit diagram of a DC motor attached to the platform shows the mechanical algorithm behind the use of the motor and the speed reducer and the way to be connected. So the motor gets its power through the electric circuitry and then it turns in a specified direction, instantly the two pulleys will rotate leading to the rotation of the solar panel.

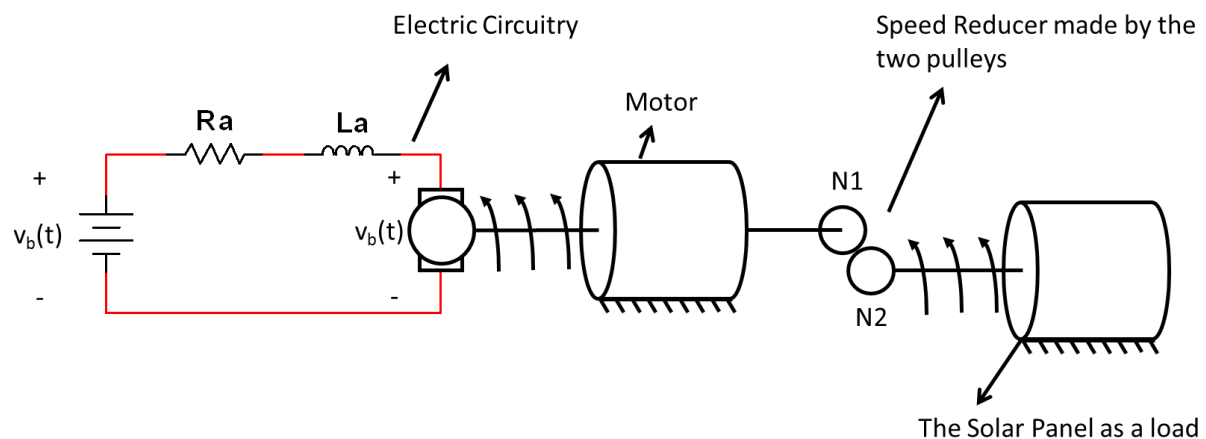


Figure 13 - Detailed circuit diagram of a DC motor attached to the platform

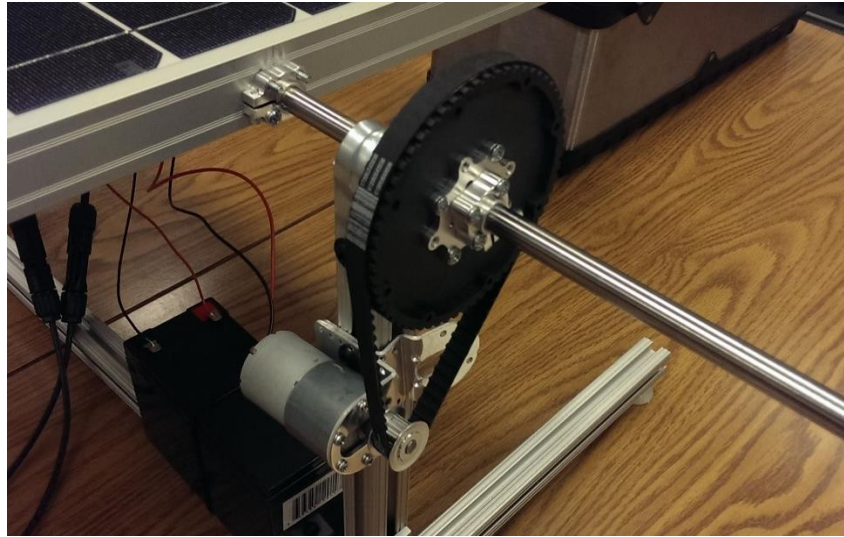


Figure 14 - Motor and Pulley Setup Side View

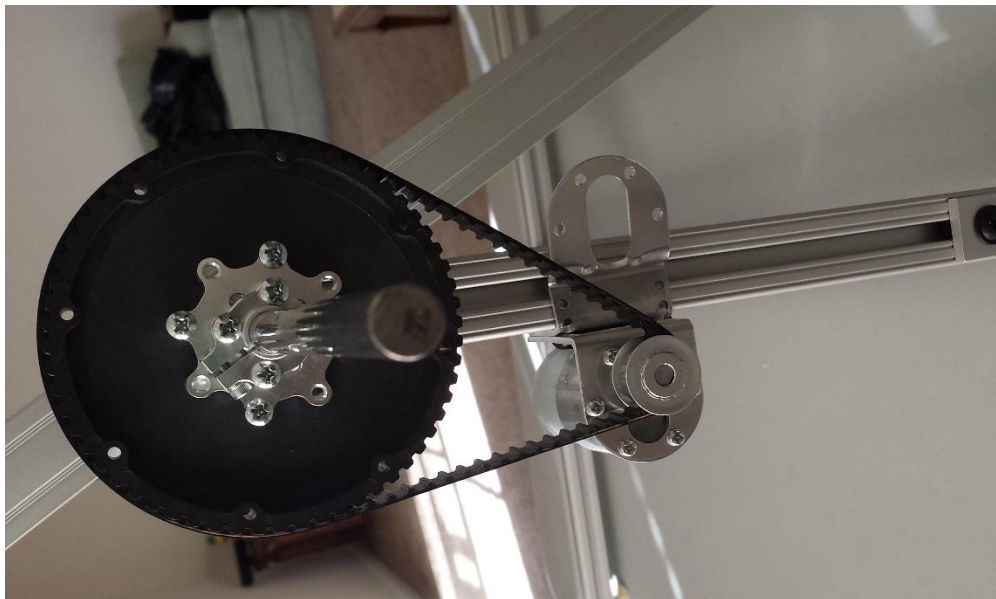


Figure 15 - Motor and Pulley Setup Front View

2.1.5 Microcontroller Software Development

Figure 16 - Microcontroller code flow chart shows a flow chart of the microcontroller software that was written by the group. The software was broken into different control functions as described below. The complete code can be found in Appendix A.

Control Loop

The control loop first updates the variables that hold the voltage values from the two photoresistors. When the system is in tracking mode, both sensor values will be checked to be sure that their values are above the low light threshold. If a low light condition was detected, the solar panel would be moved to a 90° (parallel to the Earth) angle. If the light is above the threshold, standard tracking mode would be entered.

In the standard tracking mode, the current angle of the solar panel is read from the accelerometer from one of the analog inputs of the microcontroller. This was used as the software safety; if the solar panel's current angle was moving towards an angle less than 20° or greater than 160°, then the motor would not be switched on. When the range of motion will be between 20° and 160°, the motor control function (explained later) was called with the two sensor voltages as inputs. Finally, once the two voltages are within 7.5mV of each other, the motor driver would be disabled using a motor stop function.

The control loop uses a real-time loop and will continually run as long as the microcontroller is powered.

Accelerometer Reading

The accelerometer sensor was easy to use. The output voltage would be between 0 and the supplied logic voltage (in this case 5V) depending on the angle of the sensor. The output value would be at the median voltage when the panel was at 90°. This value was then converted into an actual angle value to be used in the other programming code.

A function (gotoAngle) was used that would send the user input angle along with the current angle of the panel to the motor control function (described later). This function would loop until the desired angle was reached and then disable the motor driver.

Photoresistor Reading

The photoresistors were in a voltage divider circuit set up so that the higher the voltage between the photoresistor and standard resistor, the more light the photoresistor was detecting. These two values were both read as raw voltages into the analog inputs of the microcontroller and then used in the other programming code.

Motor Control

The motor control function was used to drive all of the inputs to the motor driver. PWM was used to control the speed of the motor based on the numerical difference between the two variables which were passed to it (either the two angles or the two photoresistor voltages). Equation 1 shows the speed scaling formula which was used. This equation always yields a

number between 0 and 75. The max value of the PWM output is 255 which would result in a 100% duty cycle. This value would result in the max speed of the DC motor being reached (80RPM) which was far too fast for this application. The 75 constant is max value which would be output to the motor and equates to about a 24% duty cycle and would only be reached when once of the comparison values was 0.

$$scale = 75 * \frac{abs(comp2 - comp1)}{\max(comp1, comp2)}$$

Equation 1

An if statement clause was added to the motor control function that ensured that the PWM scale value did not drop below 30 as the motor would not operate well below this scaling factor.

Then another if statement was used to determine which of the motor driver input pins would be grounded and which would have the PWM voltage applied to them by comparing the two input voltages and then sending an signal to the motor driver to enable it. This would determine in which the direction the motor would operate.

Mode Change

An external interrupt was set up to change a “mode” variable whenever a button was pressed within the circuitry. This variable controlled whether the system was in tracking mode or user input mode in the rest of the code.

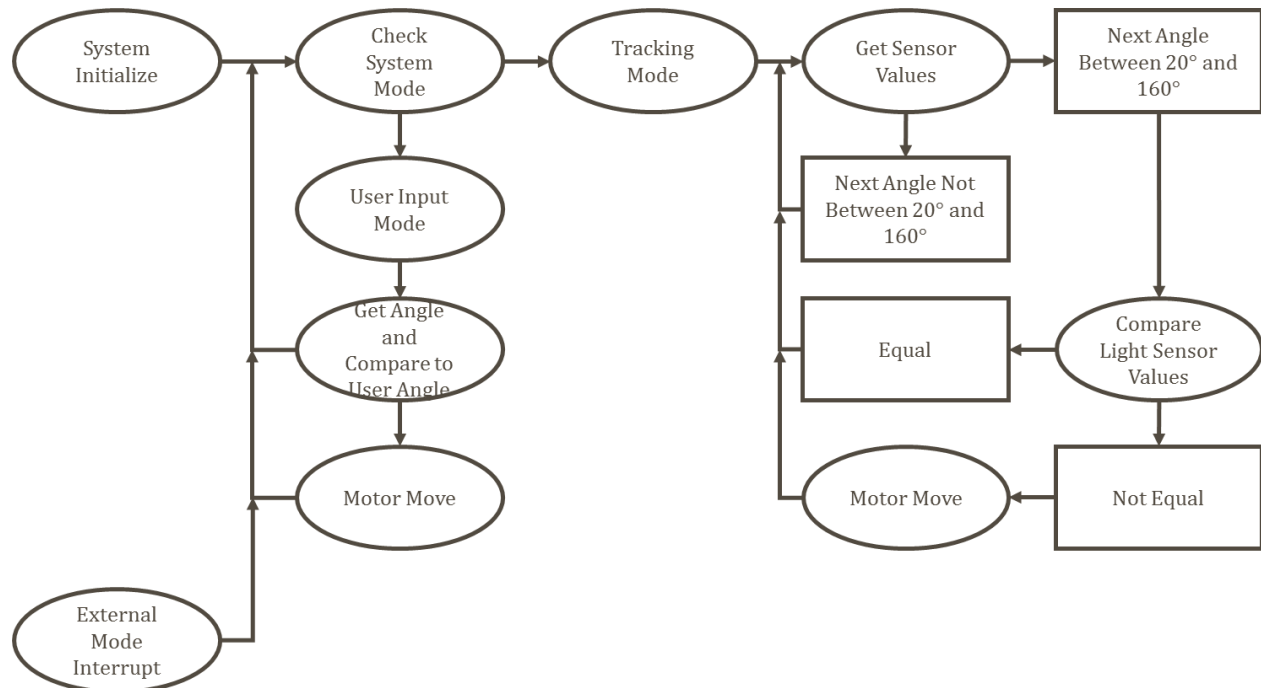


Figure 16 - Microcontroller code flow chart

2.2 Prototype Final Cost

2.2.1 Framing Material

The framing material included the entire 80/20 extruded aluminum frame, 80/20 brackets, bearings, drive shaft, and fasteners. All of the framing materials were purchased from McMaster-Cart.

2.2.2 Solar Panel and Modifications

The solar panel modifications required additional brackets and fasteners as well as the initial cost of the solar panel. There were also adaptors that were required to allow the pulleys to be attached to the drive shaft

2.2.3 Electrical Components

The electrical components included all of the sensors, microcontroller, DC motor, motor driver, and the onboard battery.

2.2.4 Overall Cost

No fabrication or assembly costs were incurred since all of the parts used were standard parts. All modifications, assembly, and coding of the software were performed by the group members and required no additional costs aside from the standard parts that were modified.

The final prototype cost was \$410.35 which was \$10.35 above the adjusted budget from the IPFW Department of Engineering of \$400. Any additional costs were absorbed by the group members. Table 1 outlines the final cost of each of the different portions of the prototype build. A Bill of Materials detailing the part descriptions, part number, suppliers, quantity, and costs of all of the parts can be found in Appendix B, Table 2.

Table 1 - Overall Cost of the Prototype Materials

Frame	\$ 95.65
Solar Panel and Modifications	\$ 148.97
Electrical Components	\$ 109.88
Shipping	\$ 55.85
	\$ 410.35

2.3 Completed Prototype

The prototype was completely assembled by the members of the team and was completed over a number of weeks due to difficulties with part orders and adaptors. The team was able to finish the initial Electrical/Programming part within two weeks, whereas due to the absence of a Mechanical Engineer and difficulties in choosing and finding the parts, the team had to spend two months to assemble the mechanical part of the project. Figure 17 shows the completed prototype.

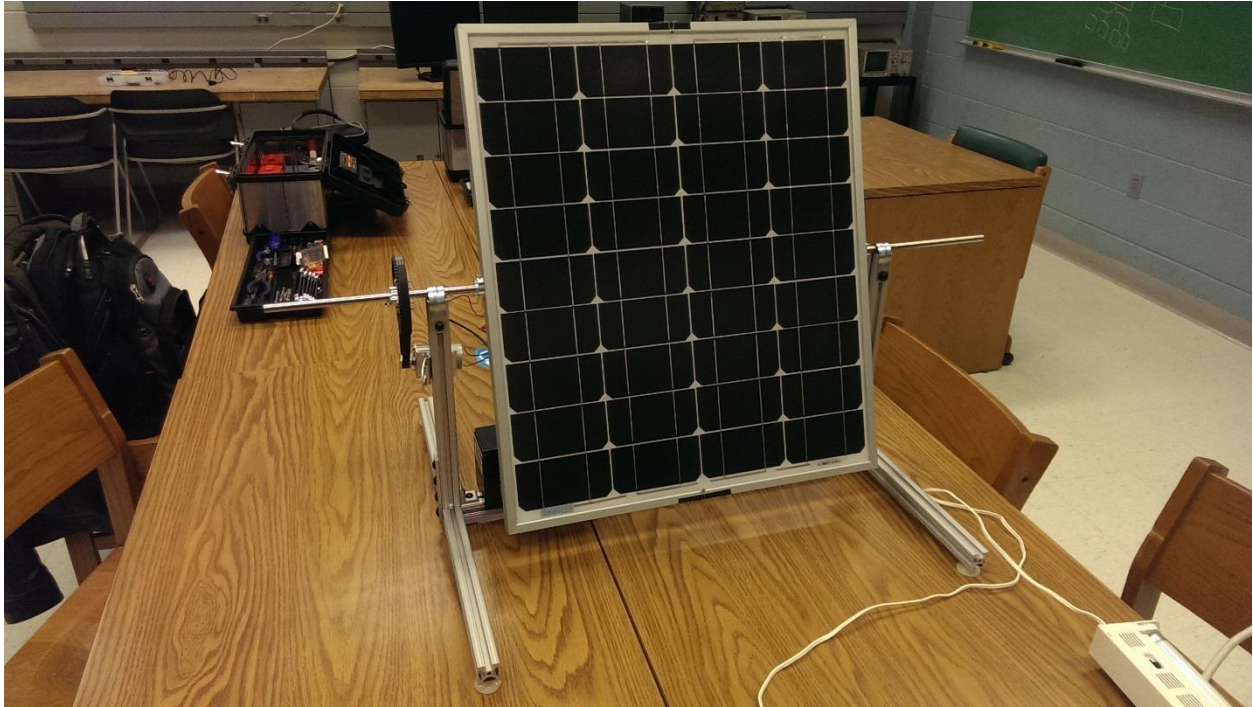


Figure 17- Prototype Completely Constructed

Section III – Testing

3.1 Testing Requirements and Methods

3.1.1 Angular Speed of Solar Platform

The first requirement that the prototype had to meet is that the platform must be able to pivot to the point with the most solar light intensity, from any position, in less than 60 seconds. In order to meet this requirement, the platform must travel an angle of 120° in 60 seconds, which is 2° a second. This requirement was tested by measuring the time it took the platform to travel from one extreme to the other in a light controlled environment (to simulate the rise of the sun when the platform is at the sunset position).

The software limited the solar panel angles of operation from 20° to 160° . The system was tested using the user input mode, telling the panel to go to 160° when it was already at the 20° position. The time to pivot from the first extreme to the second extreme was measured with a stopwatch at 15.86 seconds, which is well below the max allowable time of 60 seconds.

3.1.2 Power Consumption

The second requirement to meet is the ability of the control system to enter a power saving mode when the solar light intensity is below 25% of the solar panel's maximum power output threshold. The power saving mode must be such that the control system doesn't use more the 25% of the total on-board battery's power in 24 hours. The battery that is used is 9Ah, which constrains the control circuitry to use less than 2.25Ah of current in 24 hours or about 94mA per hour. These requirements can be tested by measuring the total current draw from the on-board battery when the control circuitry is in power saving mode.

The current draw of the control circuitry was measured using an ammeter. The max current draw of the control system was measured as 137.2mA (shown in Figure 18) at the maximum speed that the motor was limited to in the microcontroller software. When the system was idle (in power saving mode, waiting for an angle change command), all of the control circuitry drew only 60.3mA (shown in Figure 19 - Ammeter Display of Current Draw when System is Idle). This idle current is well below the required maximum draw of 94mA.



Figure 18 - Ammeter Display of Current Draw under Maximum Load



Figure 19 - Ammeter Display of Current Draw when System is Idle

3.1.3 System Accuracy

The control system must be continuously checking the position of the sun to achieve the optimum power output. This can be evaluated by using the on-board accelerometer and the microcontroller's USB connection to compare the platform's current angular position with the sun's position relative to the geographical location of the solar panel setup. The platform must be accurate to within 3° of the sun's actual position.

The maximum misalignment of the solar panel from the light source was between 0° and 2° based on the data collected during testing, full data can be found in Appendix C - Raw Solar Panel Output Data.

3.1.4 Prototype Weight

The last testing requirement to meet is the combined weight of the platform and control circuitry. This includes every component of the prototype besides the solar panel itself, which may vary. The total weight can't exceed 20 lbs. and was measured using a large scale.

The entire prototype (including the solar panel) weighed a total of 21.6 lbs. Since the solar panel itself weighs 11 lbs., the frame and control system weighs a total of 10.6 lbs. This meets the requirements outlined earlier.

3.1.5 Power Output Improvement

Power output from the solar panel was mapped using a high-side current sensor, which uses a shunt resistor to get the current through the systems. Figure 20 shows the current sensor used to collect the data.

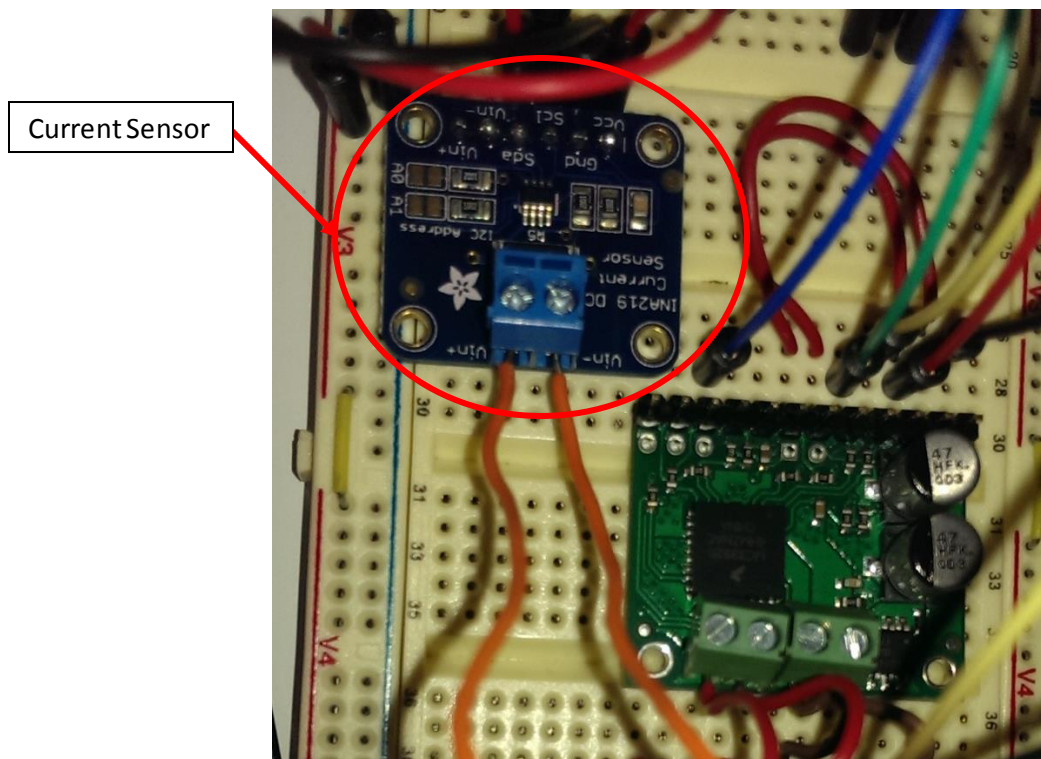


Figure 20 - Current Sensor in Control Circuitry

Figure 21-Figure 23 show the difference in power between the power outputs of the solar panel when the system was in stationary mode (set at 90°) compared to when the tracking system was on. As can be seen in the Figure 21, the power output was almost doubled when the tracking system was on and the light source was positioned at 16° . The solar panel was limited to a field of rotation from 20° to 160° which means that when the light source was 16° , the tracking system had the panel positioned at about 20° . The average increase of power due to the tracking system was around 127% compared to the output of the stationary panel at 90° .

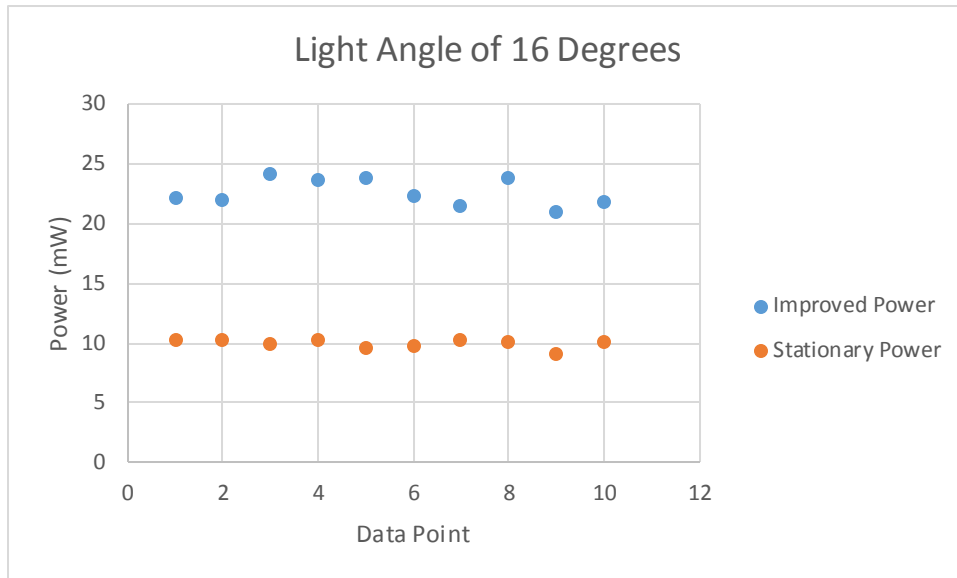


Figure 21 - Solar Panel Output of Tracking System (Light at 16 Degrees)

Figure 22 details the difference between the two systems when the light source was positioned at 45°. The average increase of power due to the tracking system was around 17% compared to the output of the stationary panel at 90°.

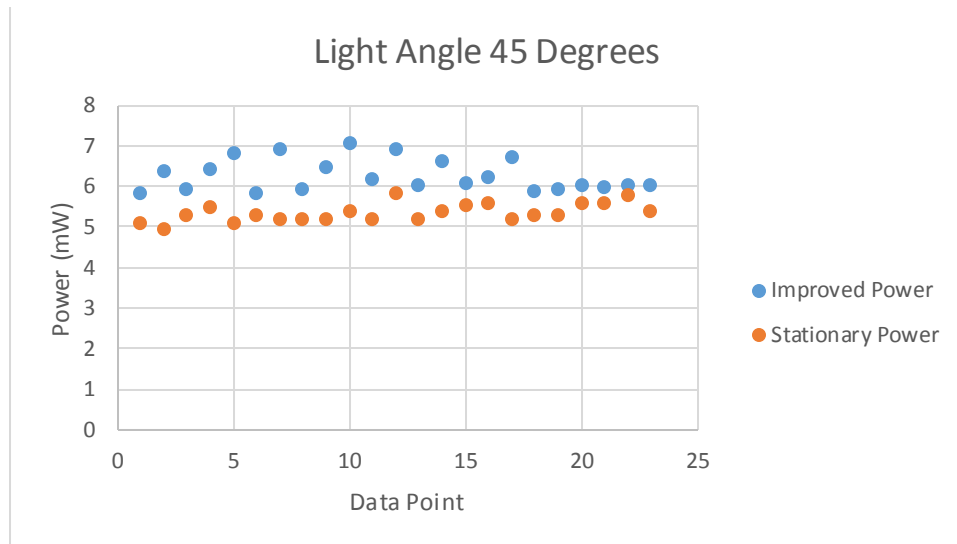


Figure 22 - Solar Panel Output of Tracking System (Light at 45 Degrees)

Finally, Figure 23 details the difference between the two systems when the light source was positioned at 90° (the highest point of the Sun). This resulted in the outputs between the two systems being almost the same as the panels position being virtually identical. Our collected data showed a 0.86% increase in power output with the tracking system on. However, this is likely an anomaly that would be closer to 0% if more data points were taken. The raw data which was collected can be found in Appendix C - Raw Solar Panel Output Data.

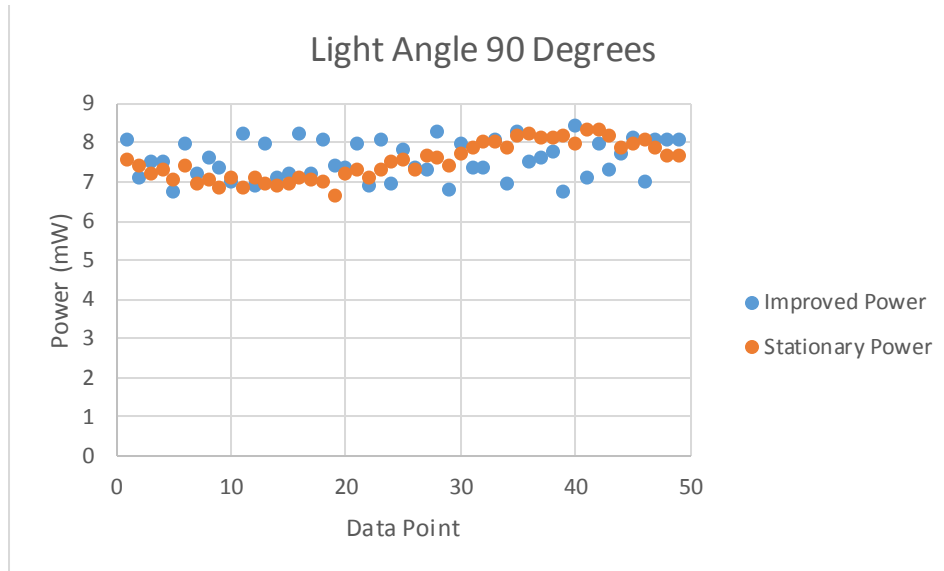


Figure 23 - Solar Panel Output of Tracking System (Light at 90 Degrees)

Section IV – Conclusions and Recommendations

4.1 Conclusions

The prototype met all of the design requirements and specifications that were outlined in the previous semester of the Capstone Senior Design Project. At the maximum speed of rotation, the solar panel was able to pivot from one extreme at 20° to the other extreme at 160° in 15.86 seconds. This meets the requirement that it travel that distance in less than 60 seconds. The control system's power consumption when it is idle is around 60mA which exceeds the maximum allowable power consumption of the system while it is in power saving mode of 94mA.

The misalignment of the solar panel to the light source was measured at a maximum of 2° , meeting the accuracy of 3° . Finally, the total of weight of the control system and solar panel frame was less than 20 lbs., measuring 10.6lbs.

Data was also taken to determine the solar panel's output power with the prototype's tracking system enabled compared to a stationary solar panel. The data was taken using a high-side current sensor that was never discussed in any of the previous presentations or reports. The use of this current sensor facilitates the capture of the data since it imports directly all of the data into a user text file. As a result of these data, the output was always greater with the tracking system on. With the biggest differences in power output (an additional 127% of the stationary solar panel's power output) appearing when the light source was at the sunrise (less than 20°) and sunset (greater than 160°).

To highlight the challenges that we faced during the building process, the team faced a lot of difficulties in figuring out the mechanical system algorithm and the way to build it so it meets the requirements. As a result of this one change in the design was taken into consideration which is the use of the 80/20 frame material that is lighter and much easier to use than the normal metal frame parts.

Overall, we are honored to know that the IPFW engineering department is fully satisfied with the prototype built, and they are keeping tabs on it for future students who are willing to develop more advanced ideas based on it.

The team would like to send special regards to the team advisor, Prof. Hossein Oloomi for monitoring each stage of this project.

4.2 Recommendations for Future Designs

Although the prototype met or exceeded all of the requirements and specifications, the team has some recommendations for future advancement of the design.

The control circuitry (microcontroller, accelerometer, motor drivers, battery, and breadboards) should be encased in a weather proof case. This will protect all of the control system from the elements, which would likely cause issues to the system during wet conditions.

It is also recommended that the photoresistors be encased in light proof material which would only allow light in from a more pointed direction. This would theoretically increase the accuracy of the system by blocking out incident light rays and focusing more on brighter light sources.

Appendix A - Source Code for Arduino

```
int motorCW = 5;
int motorCCW = 6;
int enable = 8;
int lightCCW = 12;
int lightCW = 13;
int mode = 0;

void setup(){
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);
  attachInterrupt(INT0 , modeChange, RISING);
  Serial.begin(9600);
}

void loop(){
  float voltage1, voltage2, comp, angle;

  // Retrieve the voltage readings from the light sensors.
  voltage1 = getVoltage(0);
  voltage2 = getVoltage(1) - 0.09;

  // Prints the current mode
  Serial.print("Voltage 1: ");
  Serial.print(voltage1);
  Serial.println();

  // Prints the current mode
  Serial.print("Voltage 2: ");
  Serial.print(voltage2);
  Serial.println();

  if(mode == 0){
    // Sets the panel to 90 degrees when the light sensors have low light
    if((voltage1 < 1) && (voltage2 < 1)){
      gotoAngle(90);
    } else {
```

```

// Gets the current angle of the panel.
angle = getAngle();
// Prints the current angle
Serial.print("Angle: ");
Serial.print(angle);
Serial.println();
// Compares the voltages between the two light sensors.
comp = abs(voltage1 - voltage2);
if(comp < 0.0075) {
    // Turns both LEDs on if the panel is in the correct sensor positions
    digitalWrite(lightCW, 5);
    digitalWrite(lightCCW, 5);
    // Sets the system delay to 1000ms. This is the rest time between sun tracking movements.
    motorStop();
    delay(1000);
} else if(voltage1 < voltage2) {
    digitalWrite(lightCW, 0);
    digitalWrite(lightCCW, 5);
    // Only moves the panel if is greater than 30 degrees.
    if(angle > 20){
        // Controls the motor movement using the voltage comparison between the two
        photoresistors
        motorMove(voltage1, voltage2);
    } else {
        motorStop();
    }
} else {
    digitalWrite(lightCW, 5);
    digitalWrite(lightCCW, 0);
    // Only moves the panel if is greater than 30 degrees.
    if(angle < 160){
        // Controls the motor movement using the voltage comparison between the two
        photoresistors
        motorMove(voltage1, voltage2);
    } else {
        motorStop();
    }
}

// Delays the motor movement by 5ms

```



```

    delay(5);
}
} else if(mode == 1){
    // Turns the LEDs off if in Accelerometer mode
    digitalWrite(lightCW, 0);
    digitalWrite(lightCCW, 0);
    // Goes to an accelerometer angle of 90 degrees
    gotoAngle(90);
    // Delays the motor control by 50ms
    delay(50);
}
}

float getAngle(void){
    // Reads the accelerometer voltage
    float angleVoltage = analogRead(2);
    // Converts the accelerometer voltage into an angle
    angleVoltage = (angleVoltage - 396) * 0.826;
    // Returns the panel angle
    return angleVoltage;
}

void motorStop(){
    // Clears the inputs to the motor.
    digitalWrite(motorCW, 0);
    digitalWrite(motorCCW, 0);
    digitalWrite(enable, 0);
    digitalWrite(lightCW, 0);
    digitalWrite(lightCCW, 0);
    // delay(10000);
}

void motorMove(float comp1, float comp2){
    float scale;
    float angle;
    // Gets current angle from the accelerometer
    angle = getAngle();
    // Sets the scale for the pwm controlled motor speed
    scale = 75*(abs(comp2 - comp1))/max(comp1,comp2);
    if(scale < 30){

```

```

        scale = 30;
    }
    if(comp1 < comp2){
        // Enables the motor driver
        digitalWrite(enable, 5);
        // Set the direction and speed of the motor
        digitalWrite(motorCW, 0);
        analogWrite(motorCCW, scale);
    } else {
        // Enables the motor driver
        digitalWrite(enable, 5);
        // Set the direction and speed of the motor
        analogWrite(motorCW, scale);
        digitalWrite(motorCCW, 0);
    }
    // Prints the current angle
    Serial.print("Scale: ");
    Serial.print(scale);
    Serial.println();
}

void gotoAngle(float angle){
    float currentAngle;
    // Only runs when the panel is not at the correct angle
    while(int(angle) != int(currentAngle)){
        // Retrieves the current angle from the accelerometer
        currentAngle = getAngle();
        // Doesn't tell the motor to move unless the panel position is further than 1 degree out of sync
        if(abs(currentAngle - angle) > 0.85){
            // Calls the motor control function
            motorMove(angle, currentAngle);
            // Prints the current angle
            Serial.print("Current Angle: ");
            Serial.print(currentAngle);
            Serial.println();
            // Delays the function for 50ms
            delay(50);
            if(mode == 0){
                return;
            }
        }
    }
}

```

```

    } else {
        // Calls the motor stop function
        motorStop();
    }
}

void modeChange(){
    // Changes the mode when the INT0 interrupt (Pin 2)
    mode = !mode;
}

float getVoltage(int pin)
{
    return (analogRead(pin) * 0.004992914);
    // This equation converts the 0 to 1023 value that analogRead()
    // returns, into a 0.0 to 5.0 value that is the true voltage
    // being read at that pin.
}

```

Appendix B - Detailed Bill of Materials

Table 2 - Detailed Bill of Materials

Part	Part #	Supplier	Price	Quantity	Total Price
Frame					
80/20 Frame 1'	47065T101	McMaster Carr	\$ 6.49	2	\$ 12.98
80/20 Frame 2'	47065T101	McMaster Carr	\$ 8.35	3	\$ 25.05
80/20 L-Bracket	47065T223	McMaster Carr	\$ 3.98	4	\$ 15.92
80/20 Extended Bracket	47065T173	McMaster Carr	\$ 6.05	2	\$ 12.10
Bearings	6679K15	McMaster Carr	\$ 2.90	2	\$ 5.80
10mm Drive Shaft	5448T5	McMaster Carr	\$ 23.80	1	\$ 23.80
					\$ 95.65
Solar Panel and Modifications					
Solar Panel	Renogy 50W	Amazon	\$ 86.99	1	\$ 86.99
Fasteners	Misc.	Misc.	N/A	N/A	\$ 17.90
10mm Clamping Hub	545628	ServoCity	\$ 6.79	3	\$ 20.37
6mm Bore Pinion	615384	ServoCity	\$ 5.94	1	\$ 5.94
1 inch Bore Pulley	615394	ServoCity	\$ 6.79	1	\$ 6.79
1 inch Hub Adaptor	545388	ServoCity	\$ 4.24	1	\$ 4.24
1/5" Pitch 80 Teeth Belt	160XL037	Amazon	\$ 6.74	1	\$ 6.74
					\$ 148.97
Electrical Components					
Arduino Microcontroller	UNO R3	Amazon	\$ 26.15	1	\$ 26.15
Motor Driver	MC33926	Pololu	\$ 17.95	1	\$ 17.95
Sainsmart Accelerometer	ADXL 335	Amazon	\$ 15.00	1	\$ 15.00
Photoresistors	GL5516	Amazon	\$ 0.43	2	\$ 0.86
80 RPM DC Motor	Pololu 1107	Pololu	\$ 24.95	1	\$ 24.95
12V 9Ah Battery	SLA 1090	Interstate Batteries	\$ 24.97	1	\$ 24.97
					\$ 109.88
Shipping					
Shipping	N/A	N/A	N/A	N/A	\$ 55.85
				Grand Total	\$ 410.35

Appendix C - Raw Solar Panel Output Data

Table 3 - Data Comparison between Tracking and Stationary System (Light at 16 Degrees)

	Tracking	Improved	Stationary	Percent
1	19.82	22.0964646	10.2517354	115.54%
2	19.82	21.9877505	10.2908277	113.66%
3	19.82	24.1242218	9.9722709	141.91%
4	19.82	23.6000728	10.2912151	129.32%
5	19.82	23.7921619	9.54846	149.17%
6	19.82	22.3083019	9.6906814	130.20%
7	19.82	21.4584612	10.2908277	108.52%
8	19.82	23.7937283	10.1170253	135.19%
9	19.82	21.0373067	9.1244287	130.56%
10	19.82	21.7763328	10.0794064	116.05%
				127.01%

Table 4 - Data Comparison between Tracking and Stationary System (Light at 45 Degrees)

	Angle	Improved	Stationary	Percent
1	44.6	5.7923107	5.071392	14.22%
2	44.6	6.3632397	4.9153766	29.46%
3	44.6	5.8947277	5.2828001	11.58%
4	43.78	6.4019966	5.4700045	17.04%
5	44.6	6.8133759	5.071536	34.35%
6	44.6	5.8122463	5.255496	10.59%
7	44.6	6.8935756	5.1770954	33.16%
8	44.6	5.8946161	5.1481924	14.50%
9	44.6	6.4517202	5.1771931	24.62%
10	44.6	7.0260963	5.3627495	31.02%
11	44.6	6.1369056	5.1769971	18.54%
12	44.6	6.8935756	5.7916078	19.03%
13	44.6	5.9999341	5.1769485	15.90%
14	44.6	6.5756583	5.3624496	22.62%
15	44.6	6.0353069	5.4938516	9.86%
16	44.6	6.2214279	5.57726	11.55%
17	44.6	6.6815285	5.1770458	29.06%
18	44.6	5.8724408	5.2553482	11.74%
19	44.6	5.8995203	5.2827501	11.68%
20	44.6	5.9843845	5.556252	7.71%
21	44.6	5.9543333	5.5784626	6.74%
22	44.6	6.006896	5.7699537	4.11%
23	44.6	6.0067844	5.3883543	11.48%
				17.41%

Table 5 - Data Comparison between Tracking and Stationary System (Light at 90 Degrees)

	Tracking Angle	Improved Power	Stationary Power	Percent Increase
1	88.38	8.0610923	7.5795841	6.35%
2	88.38	7.0801496	7.4295072	-4.70%
3	88.38	7.5222721	7.1856288	4.68%
4	89.21	7.5305452	7.2944283	3.24%
5	89.21	6.7377281	7.0531578	-4.47%
6	89.21	7.9549493	7.4293684	7.07%
7	89.21	7.1874914	6.9481496	3.44%
8	88.38	7.6160993	7.0536851	7.97%
9	89.21	7.3688297	6.842875	7.69%
10	89.21	6.9744834	7.1066164	-1.86%
11	89.21	8.2286062	6.8688101	19.80%
12	89.21	6.8913278	7.1066164	-3.03%
13	89.21	7.9551005	6.9482812	14.49%
14	89.21	7.0800914	6.8912	2.74%
15	89.21	7.1811747	6.92175	3.75%
16	89.21	8.2286825	7.1066827	15.79%
17	89.21	7.1876258	7.0535588	1.90%
18	89.21	8.0613203	6.9728107	15.61%
19	89.21	7.4170856	6.6575241	11.41%
20	90.03	7.3693203	7.1872911	2.53%
21	89.21	7.9848003	7.2915744	9.51%
22	89.21	6.8913278	7.1065497	-3.03%
23	89.21	8.0914154	7.2917127	10.97%
24	89.21	6.948018	7.5093894	-7.48%
25	89.21	7.7901291	7.5797996	2.77%
26	89.21	7.3462915	7.3218336	0.33%
27	89.21	7.3191757	7.6848573	-4.76%
28	89.21	8.272993	7.6164541	8.62%
29	89.21	6.81707	7.4250407	-8.19%
30	89.21	7.9849491	7.6946401	3.77%
31	89.21	7.3463611	7.8788542	-6.76%
32	89.21	7.3743753	8.0152502	-8.00%
33	89.21	8.0746059	8.0312242	0.54%
34	89.21	6.9481496	7.8789286	-11.81%
35	89.21	8.2734594	8.1673135	1.30%
36	89.21	7.5310416	8.1978807	-8.13%
37	89.21	7.6165962	8.1360511	-6.38%
38	89.21	7.7427458	8.1218156	-4.67%
39	89.21	6.7377281	8.1670055	-17.50%

40	89.21	8.4422569	7.9850249	5.73%
41	89.21	7.1066164	8.3353147	-14.74%
42	90.03	7.9850249	8.3041934	-3.84%
43	89.21	7.3186225	8.1669273	-10.39%
44	89.21	7.7142744	7.878632	-2.09%
45	89.21	8.1218156	7.9848003	1.72%
46	89.21	6.9989404	8.0608644	-13.17%
47	89.21	8.0609407	7.8785572	2.31%
48	89.21	8.0914154	7.6368961	5.95%
49	89.21	8.061016	7.6655511	5.16%
				0.86%